

Sofica Sp. z o.o.

ul. Klaudyny 16 m 136, 01-684 Warszawa

dr inż. A. Ziółkowski, tel 339796

SYSTEM ZARZADZANIA BAZA DANYCH CBASE

(Wersja 90/1)

Warszawa 1990

Spis treści:

1.Przeznaczenie i funkcje systemu.....	3
2.Struktury danych i ich reprezentacja.....	5
3.Tworzenie zbiorów.....	9
4.Wprowadzanie danych do zbiorów.....	14
5.Sortowanie zbiorów.....	20
6.Sporządzanie raportów.....	22
7.Kompresowanie zbiorów.....	25
8.Pisanie programów użytkowych.....	26

1. Przeznaczenie i funkcje systemu.

System CBase jest uniwersalnym systemem zarządzania bazą danych ze zbiorami o złożonej strukturze rekordu. Ponieważ prosta struktura rekordu występująca w relacyjnych i sieciowych bazach danych jest szczególnym przypadkiem złożonej struktury rekordu, jest oczywiste, że system CBase może być stosowany wszędzie tam gdzie są stosowane relacyjne czy sieciowe bazy danych. W wielu zastosowaniach, możliwość zdefiniowania złożonej struktury rekordu pozwala realizować systemy użytkowe w sposób znacznie prostszy niż przy wykorzystaniu standardowych systemów zarządzania bazą danych. Na przykład, w systemie obliczania płac, można zebrać wszystkie informacje mające wpływ na wynagrodzenie pracownika w jednym rekordzie. Informacja o zwolnieniach chorobowych wystąpi tylko w rekordach opisujących pracowników, którzy w danym okresie korzystali ze zwolnień. Fakt, że jeden pracownik może mieć kilka zwolnień nie powoduje tu żadnych komplikacji ponieważ wybrane grupy danych mogą być powtarzane wielokrotnie lub nie występować wcale. Wiele funkcji, których realizacja przy wykorzystaniu standardowych systemów zarządzania bazą danych jest trudna lub niemożliwa może być w prosty i naturalny sposób zrealizowana w systemie CBase.

Główne obszary zastosowań systemu CBase to:

- systemy zarządzania przedsiębiorstw,
- systemy gromadzenia i wyszukiwania informacji,
- bazy wiedzy, sztuczna inteligencja,
- przetwarzanie informacji graficznej.

Ze względu na swoje cechy, system CBase nadaje się szczególnie do realizacji sytemów użytkowych w których występuje niejednorodna informacja. Do tej grupy należą informatyczne systemy medyczne np. systemy gromadzenia informacji o pacjentach (karta choroby). Należy do niej również wiele systemów wspomagania prac inżynierskich.

System CBase pozwala tworzyć systemy zintegrowane zawierające informację tekstową, liczbową i graficzną.

System CBase składa się z uniwersalnych programów gotowych do użycia natychmiast po zdefiniowaniu struktury rekordu oraz z zestawu procedur w języku C. Programy realizują między innymi następujące funkcje:

- tworzenie zbioru o określonej strukturze rekordu,
- wprowadzanie danych do zbioru o złożonej strukturze rekordu,
- sortowanie wg dowolnie wybranego klucza,
- sporządzanie raportów z uwzględnieniem selekcji i agregacji.

Procedury systemu CBase służą do pisania programów użytkowych w języku C. Oryginalne rozwiązanie polegające na stosowaniu wskaźników do pól ustawianych automatycznie przez procedury systemu CBase na podstawie informacji o strukturze rekordu, odciąża całkowicie programistę od złożonych problemów analizy struktury rekordu. Dzięki temu rozwiązaniu programy mogą być krótkie, czytelne i przejrzyste.

2. Struktury danych i ich reprezentacja.

W systemie CBase zbiory danych składają się z rekordów zmiennej długości o złożonej strukturze. Rekordy składają się z danych.

Dopuszcza się następujące typy danych:

- łańcuchy znaków stałej długości,
- liczby całkowite krótkie (16-bitowe),
- liczby całkowite długie (32-bitowe),
- liczby zmiennoprzecinkowe (32-bitowe)
- teksty zmiennej długości,
- teksty kodowane zmiennej długości.

Dane liczbowe są przechowywane w zbiorach w formie binarnej. Teksty zmiennej długości są reprezentowane przez 16-to bitowy wskaźnik początku tekstu (numer pierwszego bajtu tekstu licząc od początku rekordu) oraz sam tekst zakończony zerowym bajtem. W podobny sposób są reprezentowane teksty kodowane zmiennej długości z tą jedynie różnicą, że same teksty są przechowywane w postaci skompresowanej.

W rekordzie mogą występować następujące struktury danych:

- struktury indeksowane - grupy danych powtarzane określoną liczbę razy w rekordzie.
- powtórzenia - grupy danych powtarzane dowolną liczbę razy w rekordzie (listy).
- opcje wykluczające się - grupy danych występujące jedynie w przypadku wybrania określonej opcji - tylko jedna z opcji może być wybrana.
- opcje sumujące się (współistniejące) - zapisane kolejno grupy

danych występujące w przypadku wybrania opcji - wiele opcji może być wybranych jednocześnie.

Dopuszcza się zagłębianie struktur danych i tak np. w opcjonalnej grupie danych może wystąpić grupa danych powtarzanych, w której występuje inna grupa danych opcjonalnych. Zarówno w przypadku opcji wykluczających się jak opcji sumujących się dopuszcza się brak danych związanych z określoną opcją. Szczególnie interesujący jest przypadek opcji wykluczających się z wszystkimi opcjami bez danych. Struktura ta odpowiada typowi danych wyliczanych (enum) języka C. Dane tego typu mogą przyjmować wartości jedynie ze skończonego zbioru. W naszym przypadku wartości są reprezentowane przez nazwy opcji.

Zarówno struktury indeksowane jak i powtórzenia są reprezentowane w rekordzie w postaci liczby 16-to bitowej określającej liczbę powtórzeń, po której występują powtarzane grupy danych.

Opcje wykluczające się są reprezentowane przez jeden bajt określający numer wybranej opcji (1-255), po którym występuje grupa danych związana z tą opcją.

Opcje sumujące się są reprezentowane przez jeden lub większą liczbę bajtów zawierających informacje o wystąpieniu opcji, po których są zapisywane grupy danych związane z wybranymi opcjami. Informacja o wybraniu opcji jest kodowana na bitach. Wybranie opcji powoduje zapalenie bitu odpowiadającemu tej opcji. W zależności od liczby możliwych do wyboru opcji rezerwowana jest potrzebna liczba bajtów.

W systemie CBase zbiory mają następującą budowę:

- informacje ogólne,
- tablice adresowe,
- tablice do szybkiego wyszukiwania,
- rekordy.

Tablice adresowe zapewniają dostęp do rekordów zbioru w wymaganej kolejności. Zawierają one adresy początku rekordu. Kolejność adresów może być ustalona przez, dostępny w systemie CBase, program sortujący. Program ten pozwala uporządkować adresy rekordów wg rosnących lub malejących wartości dowolnych danych z rekordu.

Tablice do szybkiego wyszukiwania zawierają również adresy rekordów. W systemie CBase do szybkiego wyszukiwania rekordów wg klucza stosuje się metodę kodowania mieszającego. Klucz może składać się z dowolnych danych z rekordu, które w sposób jednoznaczny identyfikują rekord. Kodowanie mieszające jest najszybszą metodą odszukiwania rekordu wg klucza, dającą czasy dostępu rzędu dziesiątych części sekundy niezależnie od wielkości zbioru. W przypadku stosowania metody drzew binarnych (B-TREE) uzyskuje się czasy 3-5 razy większe.

W zbiorze, w którym nie występują tablice adresowe i tablice do szybkiego wyszukiwania możliwy jest jedynie sekwencyjny dostęp do rekordów.

Rekordy są zapisywane w zbiorze w kolejności ich tworzenia. Kolejność ta może ulec zmianie w wyniku usuwania lub modyfikowania rekordów. Jeżeli długość rekordu po modyfikacjach jest większa od

długości pierwotnej to stary rekord jest oznaczany jako usunięty a rekord zmodyfikowany jest zapisywany na końcu zbioru. Rekordy oznaczone jako usunięte są pomijane przy przetwarzaniu. Rekordy te są ostatecznie usuwane ze zbioru przez program kompresujący.

3. Tworzenie zbiorów.

Aby utworzyć zbiór systemu CBase należy zdefiniować strukturę rekordu, określić parametry używanych tablic adresowych oraz tablic szybkiego wyszykiwania. Do tego celu używa się specjalnej notacji, której elementy przedstawiono poniżej.

Wiersze rozpoczynające się od znaku '*' są wierszami komentarza.

Wiersz rozpoczynający się od znaku ':' zawiera nazwę zbioru. Nazwa ta nie jest używana jako identyfikator i ma jedynie charakter informacyjny.

W kolejnych wierszach podawane są określenia danych i elementów struktury rekordu. Daną określa się definiując jej typ, format wydruku oraz nazwę. Oznaczenie typu i formatu jest podawane w nawiasach okrągłych bezpośrednio przed nazwą danej. Stosuje się następujące oznaczenia:

(.) - znak lub określona liczba znaków (łańcuch znaków stałej długości). Gdy nie jest to jeden znak, liczbę znaków podaje się w nawiasach kwadratowych po nazwie danej np.

(.)indeks materialu[12],

(*) - tekst dowolnej długości,

(#) - tekst kodowany dowolnej długości,

(liczba cyfr) - liczba całkowita krótka, gdy liczba cyfr mniejsza niż 5 np. (4)liczba sztuk, lub liczba całkowita długa, gdy liczba cyfr większa lub równa 5 np. (9)cena,

(liczba cyfr przed kropką . liczba cyfr po kropce) - liczba dziesiętna, pierwsza cyfra określa liczbę znaków przed

kropką a druga po kropce np. (5.2)ciężar w kg,

Nazwa danej może składać się ze znaków alfanumerycznych, spacji oraz znaków '.' '-' i '/'. Nazwę danej kończy przecinek.

Do określenia struktur danych stosuje się następującą notację:

[nazwa struktury:	- struktura indeksowana np.	[miesiąc:
grupa danych		(6.2)przychód,
]liczba powtorzen]		(6.2)rozchód,
		]12]

[nazwa powtorzenia:	- powtórzenie np.	[pozycja zlecenia:
grupa danych		(3)nr pozycji,
]...		(*)nazwa wyrobu,
		(5)ilość,
		]...

[ogólna nazwa opcji:	- opcje wykluczające się np.	[kolor:
!nazwa opcji 1:		!niebieski:
grupa danych		!czerwony:
!nazwa opcji 2:		!zielony:
grupa danych		!żółty:
...		]
]		

[ogólna nazwa opcji:	- opcje sumujące się np.	[języki:
&nazwa opcji 1:		&angielski:
grupa danych		&niemiecki:
&nazwa opcji 2:		&rosyjski:
grupa danych		&inne:
...		[język:
]		(*)nazwa,
		]...
		]

Przy zapisywaniu zagnieżdżonych struktur danych zaleca się stosowanie wcięć (charakterystycznych dla języków strukturalnych) dla zwiększenia przejrzystości.

Tablice adresowe określa się w następujący sposób:

#tadr[wielkosc tablicy]

Parametr wielkosc tablicy określa maksymalną liczbę rekordów w zbiorze. np. #tadr[1000].

Tablice adresowe do szybkiego wyszukiwania rekordów wg klucza są definiowane następująco:

#hash[wielkosc tablicy]

```
[
  :nazwa danej 1:
  :nazwa danej 2:
  ...
]
```

Wartość parametru wielkosc tablicy powinna być około 2 razy

większa od liczby rekordów w zbiorze. Zbyt mała wartość tego parametru może powodować zwiększenie czasu wyszukiwania przy dużej liczbie rekordów. Lista danych ograniczona nawiasami kwadratowymi określa klucz wyszukiwania (identyfikator rekordu) np.

```
#hash[2000]
```

```
[  
    :indeks wyrobu,  
]
```

Przykład definicji struktury rekordu:

```
***** WYROB *****
```

```
:Wyrob,
```

```
(.)indeks wyrobu[7],  
(*)nazwa,  
(*)numer rysunku,  
(*)wymiary gabarytowe,  
[wyroznik:  
    |wyrob gotowy:  
        (*)numer normy,  
        (:)data wprowadzenia normy,  
        (5.2)waga,  
        (*)charakterystyka wyrobu,  
    [elementy:  
        [rodzaj elementu:  
            |normalny:  
            |variant 1:  
            |variant 2:  
            |variant 3:  
            |variant 4:  
        ]  
        (.)indeks elementu[7],  
        (3)liczba sztuk,  
    ]...  
    |zespol:  
        [elementy:  
            [rodzaj elementu:  
                |normalny:
```

```

        |wariant 1:
        |wariant 2:
        |wariant 3:
        |wariant 4:
    ]
    (.)indeks elementu[7],
    (3)liczba sztuk,
    1...
|detal własny:
    (.)indeks materialu[12],
    (5.3)ilosc brutto na element,
    (5.3)odpad uzytkowy,
    (5.3)odpad nieuzytkowy,
    (5.3)straty w produkcji,
|detal z kooperacji:
    (.)indeks materialu[12],
    (*)producent,
    (5.3)ilosc brutto na element,
    (5.3)odpad uzytkowy,
    (5.3)odpad nieuzytkowy,
    (5.3)straty w produkcji,
]

```

```

#tadr[3000]
#hash[6000]:
[
:indeks wyrobu,
]

```

Po zdefiniowaniu struktury rekordu i określeniu tablic należy wywołać program DEF, który na podstawie tej informacji utworzy zbiór systemu CBase. Przy wywoływaniu programu DEF należy podać dwa parametry: nazwę zbioru tekstowego zawierającego definicję struktury rekordu i określenie tablic oraz nazwę tworzonego zbioru systemu CBase np.:

```
def wyrob.cbs wyrob
```

4. Wprowadzanie danych do zbioru.

W systemie CBase wprowadzanie danych do zbioru realizuje program CB. Przy wywoływaniu programu podaje się jeden parametr - nazwę zbioru do wprowadzania np.:

cb wyrob

Program zgłasza się wyświetlając pusty rekord jeżeli do zbioru nie wprowadzono jeszcze żadnych danych lub pierwszy rekord zbioru jeżeli dane były już wprowadzane.

Program CB można również wywoływać bez parametru i wówczas program zgłosi się pytaniem o nazwę zbioru.

Szczegółową instrukcję użytkowania programu wprowadzania danych można uzyskać naciskając dwa razy klawisz F1. Na ekranie pojawi się spis pozycji (ekranów) instrukcji użytkowania. Po wybraniu pozycji za pomocą klawiszy ruchu kursora (strzałek $\uparrow$ $\downarrow$ $\leftarrow$ $\rightarrow$ w prawej części klawiatury) można spowodować jej wyświetlenie po naciśnięciu klawisza Enter. Poszczególne pozycje instrukcji użytkowania można również przeglądać w kolejności ich występowania korzystając z klawiszy PgDn i PgUp. Wciśnięcie klawisza Esc kończy przeglądanie instrukcji użytkowania.

Uniwersalny program zarządzania bazą danych cb realizuje szereg funkcji umożliwiając nie tylko wprowadzanie i aktualizację zbiorów danych ale również generowanie raportów. Poniżej przedstawiono funkcje tego programu dotyczące wprowadzania danych.

----- przeglądanie zawartości zbioru -----

Po wywołaniu program wyświetla pierwszy rekord zbioru. Naciśnięcie klawiszy Ctrl PgDn powoduje wyświetlenie następnego

rekordu, natomiast klawiszy Ctrl PgUp poprzedniego rekordu. W prawej górnej części ekranu pojawia się numer aktualnie wyświetlanego rekordu a w nawiasach liczba rekordów wcześniej wprowadzonych i liczba rekordów usuniętych. Jeżeli cały rekord nie mieści się na ekranie można wyświetlić jego kolejne części naciskając klawisz PgDn. Naciskanie klawisza PgUp powoduje ponowne wyświetlanie poprzednich części rekordu a klawisza Home natychmiastowy powrót do początku rekordu. Konkretny rekord można wybrać podając jego numer (po wcześniejszym wciśnięciu klawiszy Ctrl F4) lub jego identyfikatora (po wcześniejszym wciśnięciu klawisza F4). Aby przejść na koniec zbioru należy (po wciśnięciu klawiszy Ctrl F4) podać ujemny numer rekordu np. -1. Zostanie wówczas wyświetlony pusty rekord, który po wpisaniu do niego danych będzie zapisywany na końcu zbioru.

----- wprowadzanie danych -----

Jeżeli zbiór nie zawiera żadnego rekordu, na ekranie jest wyświetlany pusty rekord, do którego można wprowadzać dane. W przypadku gdy zbiór zawiera już rekordy należy przejść najpierw na koniec zbioru w sposób podany powyżej.

Wprowadzenie danych rozpoczyna naciśnięcie klawisza » (strzałki skierowanej w prawo w prawej części klawiatury przy zgaszonym NumLock). Po wprowadzeniu jednej danej program automatycznie przechodzi do następnej. Naciśnięcie klawisza Esc lub dojście do końca rekordu kończy wprowadzanie danych. W tym momencie rekord nie jest jeszcze wprowadzony do zbioru. Dopiero naciśnięcie klawiszy Ctrl End powoduje zapisanie rekordu w zbiorze. W prawej

górnej części ekranu zmieni się podawana w nawiasach ilość rekordów w zbiorze a na ekranie zostanie wyświetlony kolejny pusty rekord do wprowadzania.

W zbiorach danych mogą występować dane różnego typu. Sposób wprowadzania danych zależy od typu:

- dane tekstowe - należy wpisać tekst i nacisnąć klawisz Enter,
- dane cyfrowe - należy wprowadzić wartość i nacisnąć klawisz Enter. (uwaga: część ułamkową zapisuje się po kropce a nie po przecinku np. -32.85).
- daty - należy wpisać cyfrowo dzień, miesiąc, rok oddzielone dowolnym znakiem i wcisnąć klawisz Enter (np. 31,12,89 lub 20:6:1989). Rok można wprowadzać podając tylko dwie ostatnie cyfry.
- opcje wyłączające się - należy wybrać jedną z opcji za pomocą klawiszy $\uparrow$ $\downarrow$ (strzałki skierowane w górę i w dół w prawej części klawiatury) i nacisnąć klawisz Enter. Wybrana opcja pojawi się jako wartość pola a wyświetlona lista opcji zniknie.
- powtórzenia - należy ponownie nacisnąć klawisz $\rightarrow$. Na ekranie pojawią się dane występujące w powtórzeniu które należy wypełnić. Po wprowadzeniu tych danych ponownie dojdzie się do powtórzeń. Można wprowadzić kolejne powtórzenie naciskając klawisz $\rightarrow$ lub zakończyć wprowadzanie powtórzeń naciskając klawisz Enter.

----- poprawianie danych -----

Aby dokonać zmian we wprowadzonych już danych należy:

- wybrać rekord, do którego będą wprowadzane poprawki, w sposób

opisany przy omawianiu przeglądania zawartości zbioru,

- wybrać zmienianą daną za pomocą klawiszy $\uparrow$ $\downarrow$.
- wprowadzić ponownie daną (nacisnąć klawisz $\rightarrow$ a następnie wpisać prawidłową wartość naciskając na końcu klawisz Enter),
- wyjść z trybu wprowadzania przez naciśnięcie klawisza Esc,
- zapisać poprawiony rekord w zbiorze naciskając klawisze Ctrl End.

W przypadku powtórzeń , po wybraniu klawiszami $\uparrow$ $\downarrow$ odpowiedniego powtórzenia można:

- wpisać dodatkowe powtórzenie na końcu listy powtórzeń - należy ponownie nacisnąć klawisz $\rightarrow$.
- wpisać dodatkowe powtórzenie przed wybranym powtórzeniem - należy nacisnąć klawisz Ins.
- usunąć wybrane powtórzenie - należy nacisnąć klawisz Del.

----- usuwanie rekordu -----

Aby usunąć rekord należy:

- wybrać usuwany rekord w sposób opisany przy omawianiu przeglądania zawartości zbioru,
- usunąć rekord ze zbioru przez naciśnięcie klawiszy Ctrl F5.

W prawej części ekranu zostanie zwiększona liczba rekordów usuniętych (podana w nawiasach po znaku $\diagup$) Rekord nie jest wymazywany całkowicie ze zbioru a jedynie oznaczany jako usunięty. Rekordy oznaczone jako usunięte są pomijane przy przetwarzaniu. Usunięcie rekordu nie zmienia wielkości zbioru. Dopiero

kompresowanie zbioru (wykonywane osobnym programem o nazwie KOM) usuwa całkowicie wszystkie rekordy oznaczone jako usunięte i zmniejsza wielkość zbioru. Kompresowanie zbioru zawierającego wiele rekordów może trwać dość długo i dlatego nie warto go robić zbyt często. Program kompresowania KOM wywołuje się przez podanie jego nazwy. Jedynym parametrem jaki należy podać po wywołaniu programu jest nazwa kompresowanego zbioru. Dopuki zbiór nie zostanie skompresowany usunięte rekordy mogą być ponownie włączone do zbioru przez naciśnięcie klawiszy Ctrl F5.

----- definiowanie wzorca rekordu -----

Pewne dane mogą mieć taką samą wartość w kolejnych rekordach np. data wystawienia dokumentu w dokumentach z tego samego dnia. Aby uniknąć wielokrotnego wprowadzania tych samych danych można ustawić wzorzec rekordu przez wprowadzenie danych wzorcowych do pustego rekordu i zapisanie wzorca przez naciśnięcie klawiszy Ctrl Home. Przy wprowadzaniu nowych rekordów, zamiast pustego rekordu będzie się pojawiał rekord z danymi wzorcowymi, które nie muszą już być wprowadzane.

----- selekcja informacji -----

Ustawienie warunków selekcji, realizowane w programie CB przez wciśnięcie klawisza F3. Ustawienie warunków selekcji wpływa na przeglądanie zawartości zbioru (Ctrl PgUp, Ctrl PgDn) - rekordy nie spełniające warunków selekcji są pomijane. Rekordy nie spełniające warunków selekcji są również pomijane przy sporządzaniu raportów.

Po wybraniu opcji ustawiania warunków selekcji program CB

wyświetla na ekranie listę danych w zbiorze oraz w górnej części ekranu nawias otwierający - początek warunków selekcji. Możliwe jest definiowanie złożonych warunków selekcji za pomocą operatorów "i" (klawisz &), oraz "lub" (klawisz |) oraz nawiasów.

Najprostszym elementem warunku selekcji jest relacja. Aby określić relację należy, korzystając z klawiszy ruchu kursora (←, →, PgUp, PgDn) i klawisza Enter, wybrać daną będącą pierwszym argumentem relacji. Następnie należy określić operator relacji oraz drugi argument. Drugim argumentem relacji może być inna dana lub wartość. Wartość można wprowadzić po wciśnięciu klawisza → (ruch kursora w prawo). Relacje mogą być łączone za pomocą operatorów logicznych & |. Nawiasy () pozwalają budować złożone warunki selekcji. Nawias otwierający może wystąpić bezpośrednio po operatorze logicznym lub innym nawiasie otwierającym.

Definiowane warunki selekcji są wyświetlane w górnej części ekranu w specjalnej, przejrzystej postaci. Klawisz ⌫ (backspace) pozwala skasować zdefiniowany ostatnio warunek. Wciśnięcie klawisza Enter po zdefiniowaniu poprawnych warunków selekcji (z prawidłową liczbą nawiasów) powoduje ich zapamiętanie i powrót do innych funkcji. Po ponownym wciśnięciu klawisza F3 zostaną wyświetlone ostatnio zdefiniowane warunki selekcji, które mogą być następnie modyfikowane. Wciśnięcie klawisza Esc przed zapamiętaniem warunków selekcji powoduje ich wykasowanie. Gdy zapamiętane warunki są wyświetlane ponownie, wciśnięcie klawisza Esc powoduje jedynie przejście do innych funkcji. Wciśnięcie klawisza Home zawsze kasuje wszystkie warunki selekcji.

5. Sortowanie zbiorów.

Porządkowanie rekordów wg dowolnego klucza realizuje program SOR. Program SOR uruchamia się podając jego nazwę. Po zgłoszeniu się programu należy wprowadzić nazwę sortowanego zbioru. Program wyświetli listę danych w zbiorze podając dodatkowo ich standardowe długości. Korzystając z klawiszy ruchu kursora (← → PgUp PgDn) i klawisza Enter należy wybrać dane, wg których ma być porządkowany zbiór, i dołączyć je do listy indeksów sortowania. Indeksy sortowania są wyświetlane w górnej części ekranu. Aby dołączyć wybraną daną do listy indeksów sortowania należy wcisnąć klawisz + lub Ins jeżeli zbiór ma być porządkowany wg rosnących wartości tej danej lub klawisz - jeżeli zbiór ma być porządkowany wg malejących wartości wybranej danej. Naciśnięcie klawisza Del usuwa wybraną daną z listy indeksów sortowania. Kolejność danych na liście indeksów sortowania decyduje o kolejności rekordów w zbiorze ponieważ wartości indeksów sortowania są porównywane zgodnie z kolejnością na liście, tak więc wartość dalszego indeksu ma znaczenie tylko wtedy gdy w porównywanych rekordach wszystkie wartości poprzedzających go indeksów są w obu rekordach takie same.

Na ekranie oprócz nazw danych są wyświetlane ich standardowe długości. Wielkości te są istotne jedynie w przypadku danych tekstowych, określają bowiem ile znaków jest branych pod uwagę w porównaniach. Jeżeli wartości tych danych w różnych rekordach różnią się jedynie na dalszych pozycjach, wówczas kolejność tych rekordów po sortowaniu zostanie ustalona w sposób przypadkowy.

Standardowe długości indeksów sortowania mogą być zmienione. Aby zmienić długość indeksu należy nacisnąć klawisz → (ruch kursora w prawo) i następnie wprowadzić nową wartość. Zwiększanie długości indeksów sortowania wydłuża w pewnym stopniu czas sortowania.

Po ustaleniu listy indeksów sortowania, naciśnięcie klawisza Enter rozpoczyna sortowanie. Po skończeniu sortowania program wraca do pytania o nazwę zbioru do sortowania. Można wówczas wprowadzić nazwę kolejnego zbioru do sortowania lub zakończyć pracę naciskając klawisz Esc.

Program sortujący nie zmienia kolejności rekordów w zbiorze a jedynie ustala nowe adresy w tablicy adresowej. Dlatego na czas sortowania ma wpływ raczej długość indeksów sortowania niż długość rekordów w zbiorze. Oczywiście czas sortowania zależy od ilości rekordów w zbiorze.

6. Sporządzanie raportów.

Raporty w systemie CBase uzyskuje się za pomocą uniwersalnego programu wprowadzania danych CB. Program ten oprócz funkcji wprowadzania danych opisanych w rozdz. 4 umożliwia drukowanie raportów na drukarce lub wyświetlanie na ekranie monitora.

Aby sporządzić raport należy po uruchomieniu programu CB i wybraniu zbioru wcisnąć:

F6 - jeżeli raport ma być wyświetlany na ekranie,

F7 - jeżeli raport ma być wydrukowany na drukarce.

Program wyświetli na ekranie listę danych występujących w zbiorze podając dla każdego standardową szerokość kolumny, w której będą drukowane wartości tej danej, oraz numer znaku w wierszu od którego zaczyna się ta kolumna. Ponieważ szerokość drukarki jest ograniczona, ilość kolumn jest też ograniczona. Numer znaku w wierszu, od którego się zaczyna kolumna jest podawany jedynie dla danych wybranych do drukowania. Program wstępnie wybiera do drukowania dane w kolejności w jakiej występują w zbiorze ale tylko tyle ile może być wydrukowanych. Użytkownik może zmienić listę danych wybranych do drukowania korzystając z następujących klawiszy:

Home - usuwa wszystkie dane z listy danych do drukowania,

↑ * PgUp PgDn - umożliwia wybranie danej,

Ins - dołącza wybraną daną na końcu listy danych do drukowania,

Del - usuwa wybraną daną z listy danych do drukowania.

Kolejność w jakiej dane są dołączane do listy danych do drukowania decyduje o kolejności danych na wydruku.

Na ekranie obok nazwy danej jest podawana szerokość kolumny, w której będą drukowane wartości tej danej. Szerokość kolumny może być zmieniana. Aby zmienić szerokość kolumny należy wcisnąć klawisz > (ruch kursora w prawo) a następnie wprowadzić nową wartość. Szerokość danych zawierających teksty zmiennej długości może być ustalana dowolnie. Gdy tekst, ze względu na zbyt małą szerokość kolumny, nie mieści się w jednym wierszu, może być podzielony na dwie lub więcej części drukowane w kolejnych wierszach. Ustawienie zbyt wąskiej kolumny może powodować wydłużenie całego wydruku. Dane numeryczne są drukowane zawsze w jednym wierszu. W przypadku gdy w rekordzie występują powtórzenia, kolejne pozycje powtórzenia są drukowane w kolejnych wierszach.

Dane numeryczne mogą być określane jako dane sumowane. Aby określić daną jako sumowaną należy po wybraniu danej wcisnąć klawisz + . Dla danych sumowanych jest obliczana suma końcowa z wszystkich rekordów zbioru ale także mogą być obliczane sumy częściowe o ile zostaną określone tzw. indeksy sumowania. Aby określić indeks sumowania należy po wybraniu danej wcisnąć klawisz * . Można określić maksymalnie 3 różne indeksy sumowania. Jeżeli są określone indeksy sumowania to dla wszystkich danych sumowanych są drukowane sumy częściowe w momencie zmiany wartości indeksu sumowania. Drukowanie raportu z sumami częściowymi ma tylko wtedy sens gdy zbiór jest uporządkowany wg indeksów sumowania. Kolejność indeksów sumowania jest istotna. Zmiana wartości indeksu sumowania określonego wcześniej powoduje również wydrukowanie sum częściowych indeksów określonych później niezależnie od tego czy

ich wartość uległa zmianie czy nie.

Ustawienie warunków selekcji (patrz rozdział 4) powoduje, że w raporcie są uwzględniane jedynie rekordy spełniające te warunki. Warunki selekcji odnoszą się zawsze do całego rekordu. Nie można ustawić warunków selekcji dla powtórzeń występujących w rekordzie.

W przypadku gdy raport jest wyświetlany a nie drukowany, na ekranie pojawia się jedynie jego część. Korzystając z klawiszy ruchu kursora (↑ ↓ PgUp PgDn ← → Home End) można przeglądać kolejne jego części. Wciśnięcie klawisza Esc kończy przeglądanie.

7. Kompresowanie zbioru.

W przypadku usuwania lub modyfikowania rekordów w zbiorze mogą się pojawić niewykorzystane miejsca. Podczas usuwania rekordy są jedynie oznaczane jako usunięte a miejsce zajmowane przez rekord pozostaje niewykorzystane. Jeżeli długość zmodyfikowanego rekordu jest większa od pierwotnej długości tego rekordu to stary rekord zostaje oznaczony jako usunięty a nowy zapisany na końcu zbioru. Również w tym przypadku powstaje niewykorzystane miejsce. Podczas przetwarzania rekordy oznaczone jako usunięte są pomijane, tak więc istnienie niewykorzystanych miejsc nie ma wpływu na działanie systemów użytkowych. Zbiory zajmują wtedy jednak niepotrzebnie dużo miejsca. Do usuwania ze zbioru rekordów oznaczonych jako usunięte służy program KOM. Program ten uruchamia się podając jego nazwę z ewentualnie jednym parametrem - nazwą kompresowanego zbioru. Jeżeli program zostanie wywołany bez parametru to wówczas zgłasza się pytaniem o nazwę zbioru do skompresowania.

Po skompresowaniu jednego zbioru program zgłasza się ponownie pytaniem o nazwę zbioru. Można wówczas albo podać nazwę następnego zbioru do kompresowania lub zakończyć pracę programu KOM wciskając klawisz Esc.

Program KOM odtwarza tablice adresowe i tablice do szybkiego wyszukiwania rekordów wg klucza.

8. Pisanie programów użytkowych.

Oprócz gotowych do użycia programów w skład systemu CBase wchodzi biblioteka procedur ułatwiających pisanie programów użytkowych w języku C.

Aby użyć procedur z biblioteki należy w programie użytkowym umieścić wiersz:

```
#include <\cbase\bib\bib.h>
```

,który określa nazwę zbioru zawierającego prototypy procedur (funkcji) systemu CBase oraz definicje typów zmiennych używanych w procedurach.

Dla każdego zbioru CBase używanego w programie należy zadeklarować strukturę typu DBF np.:

```
DBF zb1,zb2;
```

Struktura ta zawiera parametry używane przez procedury.

Dla każdej używanej danej z rekordu zbioru CBase należy zadeklarować wskaźnik do zmiennej tego samego typu co dana. Dla danych, które nie są używane w programie choć występują w rekordzie można nie definiować wskaźników. Poszczególnym elementom definicji struktury rekordu odpowiadają następujące typy:

Dane znakowe:

```
(.)nazwa danej,          --->      char *nazwa danej;  
(.)nazwa danej[i],      --->      char *nazwa danej[i];
```

gdzie i - liczba znaków, i>0.

Dane tekstowe:

(*)nazwa danej, ---> CBTEXT nazwa danej;

(#)nazwa danej, ---> CBTEXT nazwa danej;

gdzie typ CBTEXT jest strukturą zawierającą dwa składniki:

int *i - wskaźnik do liczby całkowitej określający położenie
tekstu od początku rekordu,

char *c - wskaźnik tekstu.

Liczby całkowite krótkie:

(i)nazwa danej, ---> int *nazwa danej;

gdzie i - liczba znaków, $0 < i < 5$.

Liczby całkowite długie:

(i)nazwa danej, ---> long *nazwa danej;

gdzie i - liczba znaków, $i > 4$.

Liczby dziesiętne:

(i.j)nazwa danej, ---> float *nazwa danej;

gdzie i - liczba znaków przed kropką, j - liczba znaków po kropce.

Powtórzenia:

[nazwa powtorzenia: ---> unsigned int *nazwa powtorzenia;

...

]....

Opcje wykluczające się:

```
[nazwa opcji:      --->      unsigned char *nazwa opcji;
```

```
!opcja i:
```

```
...
```

```
]
```

Opcje sumujące się:

```
[nazwa opcji:      --->      unsigned char *nazwa opcji[i];
```

```
&opcja i:
```

```
...
```

```
]
```

gdzie $i = (\text{liczba opcji} - 1) / 8 + 1$

Ponieważ w systemie CBase używa się długich nazw danych, często złożonych z kilku słów, dlatego dla oznaczenia odpowiadających im wskaźników wygodnie jest używać skrótów nazw danych a nie pełnych nazw. Nazwy te mogą bowiem występować wielokrotnie w programie i ich pisanie może być zbyt uciążliwe. Nazwy wskaźników mogą być wybrane przez programistę w sposób całkowicie dowolny, jednak stosowanie jednolitych zasad tworzenia tych nazw bardzo ułatwia uruchamianie programów. Ponieważ te same nazwy danych mogą występować w różnych zbiorach dlatego przy tworzeniu wskaźników warto również do nazwy wskaźnika dołączać skrót nazwy zbioru , z którego dana pochodzi.

Procedury biblioteczne systemu CBase można podzielić na trzy grupy:

- procedury przygotowawcze,
- procedury dotyczące całego rekordu,
- procedury dotyczące elementów struktury rekordu.

Do pierwszej grupy wchodzi następujące procedury:

----- opendir -----

Otwarcie zbioru.

```
void opendir(DBF *dbf, char *filename, unsigned int MAXREK);
```

Procedura odszukuje zbiór o nazwie *filename*, wczytuje część parametrów do struktury *dbf* oraz ustawia wartości początkowe pozostałych. Parametr *MAXREK* określa wielkość bufora rekordu w pamięci operacyjnej. Procedura musi być wywołana przed wywołaniem jakiegokolwiek innej procedury odnoszącej się do zbioru.

Przykład:

```
#include <\cbase\bib\bib.h>
```

```
DBF wyrob;
```

```
opendir(&wyrob, "\\tpp\\wyrob", 1000);
```

----- assfield -----

Deklaracja wskaźnika do elementu struktury rekordu (danej).

```
void assfield(DBF *dbf, void far *field, char *fieldname);
```

Procedura przyporządkowuje elementom struktury rekordu *fieldname* zbioru *dbf* zadeklarowany w programie wskaźnik do zmiennej tego samego typu. Parametr *field* jest adresem zadeklarowanego wskaźnika. Przy rekordach o złożonej wielopoziomowej strukturze,

przy określaniu elementu struktury należy podać nazwy elementów struktury od poziomu najwyższego do najniższego oddzielone znakiem dwukropka. Procedura `assfield` jest wywoływana po procedurze `opendbf` ale przed innymi procedurami.

Przykład:

```
char *iw[7];
CBTEXT nw;
char *iew[7];
...
assfield(&wyrob,&iw,"indeks wyrobu");
assfield(&wyrob,&nw,"nazwa");
assfield(&wyrob,&iew,"wyroznik:wyrob gotowy:indeks elementu");
```

----- tadrno -----

Wybór tablicy adresowej.

```
void tadrno(DBF *dbf, unsigned char tabno);
```

Procedura służy do wybrania tablicy adresowej o numerze `tabno`. Jeden zbiór może mieć kilka tablic adresowych określających inną kolejność rekordów w zbiorze. Procedura `opendbf` standardowo wybiera pierwszą tablicę adresową. Procedura `tabno` pozwala zmienić ten standardowy wybór.

Przykład:

```
tadrno(&pracown,2);
```

----- hashno -----

Wybór tablicy szybkiego wyszukiwania rekordów wg identyfikatora.

```
void hashno(DBF *dbf, unsigned char tabno);
```

Procedura służy do wyboru tablicy szybkiego wyszukiwania rekordów wg identyfikatora (klucza) o numerze *tabno*. Jeden zbiór może mieć kilka tablic szybkiego wyszukiwania dla różnych identyfikatorów. Procedura *opendbf* standardowo wybiera pierwszą tablicę wyszukiwania. Procedura *hashno* pozwala zmienić ten standardowy wybór.

Przykład:

```
hashno(&procown,2);
```

W bibliotece systemu CBase występują następujące procedury dotyczące całego rekordu:

----- firstrec -----

Przejdź do pierwszego rekordu.

```
void firstrec(DBF *dbf);
```

Procedura ustawia wskaźnik aktualnego rekordu na pierwszy rekord zbioru.

Przykład:

```
firstrec(&wyrob);
```

----- lastrec -----

Przejdź do ostatniego rekordu.

```
void lastrec(DBF *dbf);
```

Procedura ustawia wskaźnik aktualnego rekordu na koniec zbioru.

Używana do dopisywania rekordów w zbiorach sekwencyjnych.

Przykład:

```
lastrec(&wyrob);
```

----- nextrec -----

Przejdź do następnego rekordu.

```
int nextrec(DBF *dbf);
```

Procedura ustawia wskaźnik aktualnego rekordu na następny rekord i zwraca wartość 1. Gdy koniec zbioru zwraca wartość 0.

Przykład:

```
nextrec(&plan);
```

----- findrecno -----

Przejdź do rekordu o podanym numerze.

```
int findrecno(DBF *dbf, unsigned long recno);
```

Procedura ustawia wskaźnik aktualnego rekordu na rekord o numerze *recno*. Zwraca wartość 1 gdy rekord o tym numerze istnieje a 0 gdy nie.

Przykład:

```
findrec(&wyrob,100);
```

----- findrec -----

Odszukaj rekord o podanym identyfikatorze.

```
int findrec(DBF *dbf, char *key);
```

Procedura odszukuje rekord o podanym identyfikatorze (wskazywanym przez parametr *key*). Zwraca wartość 1 gdy rekord o tym identyfikatorze istnieje a 0 gdy nie.

Przykład:

```
CBTEXT iwp;
```

```
findrec(&wyrob,iwp.c);
```

----- newrec -----

Utworzenie nowego rekordu w pamięci operacyjnej.

```
void newrec(DBF &dbf);
```

Procedura tworzy pusty rekord i wstępnie ustawia wskaźniki do elementów struktury rekordu (zadeklarowane procedurą assfield).

Po wpisaniu danych do rekordu, bezpośrednio lub za pomocą procedur dotyczących elementów struktury rekordu, rekord może być wpisany do zbioru procedurą writerec.

Przykład:

```
newrec(&faktura);
```

----- getrec -----

Wczytanie rekordu bez ustawiania wskaźników danych.

```
void getrec(DBF *dbf);
```

Procedura wczytuje aktualny rekord bez ustawiania wskaźników do elementów struktury rekordu. Programy pisane z wykorzystaniem procedury getrec mogą działać szybciej niż programy wykorzystujące uniwersalną procedurę readrec, ale programista musi pisać dodatkowe instrukcje interpretacji danych rekordu. Wskaźnik początku rekordu zawiera dbf.poczrek.

Przykład:

```
long ilosc;
```

```
DBF plan;
```

```

do
{
    getrec(&plan);
    ilosc=*(long *) (plan.poczrek+7);
    ...
}
while(nextrec(&plan));

```

----- readrec -----

Wczytanie aktualnego rekordu i ustawienie wskaźników danych.

```
void readrec(DBF *dbf);
```

Procedura wczytuje aktualny rekord oraz ustawia wskaźniki do elementów struktury rekordu. Gdy w rekordzie występują powtórzenia, wskaźniki z grupy danych powtarzanych są ustawiane na pierwsze powtórzenie.

Przykład:

```

long ilosc,*ilp;
DBF plan;
...
assfield(&plan,&ilp,"ilosc planowana");
...
do
{
    readrec(&plan);
    ilosc=*ilp;

```

```
}
```

```
while(nextrec(&plan));
```

```
----- writerec -----
```

Wpisanie rekordu z pamięci operacyjnej do zbioru.

```
void writerec(DBF *dbf);
```

Procedura aktualizuje rekord w zbiorze lub dopisuje nowy rekord gdy rekordu nie ma w zbiorze.

Przykład:

```
long *ilp, *cenap, *wartp;
```

```
DBF plan;
```

```
...
```

```
assfield(&plan,&ilp,"ilosc planowana");
```

```
assfield(&plan,&cenap,"cena");
```

```
assfield(&plan,&wartp,"wartosc");
```

```
...
```

```
do
```

```
{
```

```
readrec(&plan);
```

```
*wart=*ilp * (*cenap);
```

```
writerec(&plan);
```

```
}
```

```
while(nextrec(&plan));
```

```
----- delrec -----
```

Usunięcie aktualnego rekordu ze zbioru.

```
void delrec(DBF *dbf);
```

Procedura oznacza aktualny rekord jako usunięty.

Przykład:

```
findrec(&wyrob, "0010000");
```

```
delrec(&wyrob);
```

Procedury dotyczące elementów struktury działają jedynie na rekordzie znajdującym się w pamięci operacyjnej. Przed użyciem procedur z tej grupy należy wczytać rekord ze zbioru procedurą readrec lub utworzyć rekord procedurą newrec. Do grupy procedur dotyczących elementów struktury należą następujące programy:

----- nextrep -----

Następne powtórzenie.

```
int nextrep(DBF *dbf, unsigned int *rep);
```

Procedura ustawia wskaźniki danych powtarzanych na następne powtórzenie. Parametr rep jest wskaźnikiem powtórzenia deklarowanym procedurą assfield. Parametr ten wskazuje na aktualną liczbę powtórzeń w rekordzie. Procedura zwraca aktualny numer powtórzenia lub liczbę 0 gdy nie ma więcej powtórzeń.

Przykład:

Sumowanie wartości danej "ilosc" występującej w powtórzeniu "pozycja".

```
DBF zamow;
```

```
unsigned int *pozycja;
```

```
float *ilosc,suma;
```

...

```
readrec(&zamow);
```

```
suma=0;
```

```
do
```

```
    suma+=*ilosc;
```

```
while(nextrep(&zamow,pozycja));
```

----- prevrep -----

Poprzednie powtórzenie.

```
int prevrep(DBF *dbf, unsigned int *rep);
```

Procedura ustawia wskaźniki danych powtarzanych na poprzednie powtórzenie. Parametr *rep* jest wskaźnikiem powtórzenia deklarowanym procedurą *assfield*. Parametr ten wskazuje na aktualną liczbę powtórzeń w rekordzie. Procedura zwraca aktualny numer powtórzenia lub 0 gdy nie ma więcej powtórzeń.

----- firstrep -----

Pierwsze powtórzenie.

```
int firstrep(DBF *dbf, unsigned int *rep);
```

Procedura ustawia wskaźniki danych powtarzanych na pierwsze powtórzenie. Parametr *rep* jest wskaźnikiem powtórzenia deklarowanym procedurą *assfield*. Parametr ten wskazuje na aktualną liczbę powtórzeń w rekordzie. Procedura zwraca 0 gdy nie ma powtórzeń lub 1 gdy jest choć jedno powtórzenie.

----- lastrep -----

Ostatnie powtórzenie.

```
int lastrep(DBF *dbf, unsigned int *rep);
```

Procedura ustawia wskaźniki danych powtarzanych na ostatnie powtórzenie. Parametr *rep* jest wskaźnikiem powtórzenia deklarowanym procedurą *assfield*. Parametr ten wskazuje na aktualną liczbę powtórzeń w rekordzie. Procedura zwraca 0 gdy nie ma powtórzeń lub 1 gdy jest choć jedno powtórzenie.

----- findrep -----

Wybrenie powtórzenia o podanym numerze.

```
int findrep(DBF *dbf, unsigned int *rep, unsigned int repno);
```

Procedura ustawia wskaźniki danych powtarzanych na powtórzenie o numerze *repno*. Parametr *rep* jest wskaźnikiem powtórzenia deklarowanym procedurą *assfield*. Parametr ten wskazuje na aktualną liczbę powtórzeń w rekordzie. Procedura zwraca 1 gdy jest powtórzenie o numerze *repno*, 0 - gdy nie ma.

Przykład:

```
findrep(&zamow,pozycja,3);
```

----- insrep -----

Dopisanie powtórzenia przed aktualnie wybranym powtórzeniem.

```
int insrep(DBF *dbf, unsigned int *rep);
```

Procedura dopisuje puste powtórzenie przed aktualnie wybranym i ustawia wskaźniki danych powtarzanych na dopisane powtórzenie. Procedura zawsze zwraca 1.

Przykład:

Dopisanie pierwszego powtórzenia i podstawienie wartości zmiennej "suma" na daną "ilosc".

```
DBF zamow;  
...  
unsigned int *pozycja;  
float suma, *ilosc;  
...  
readrec(&zamow);  
insrep(&zamow, pozycja);  
*ilosc=suma;  
writerec(&zamow);
```

----- apprep -----

Dopisanie powtórzenia po aktualnie wybranym powtórzeniu.

```
int apprep(DBF *dbf, unsigned int *rep);
```

Procedura dopisuje puste powtórzenie po aktualnie wybranym powtórzeniu i ustawia wskaźniki danych powtarzanych na dopisane powtórzenie. Procedura zwraca zawsze 1..

Przykład:

Dopisanie nowego powtórzenia do listy powtórzeń.

```
DBF zamow;  
...  
unsigned int *pozycja;  
...  
readrec(&zamow);  
lastrep(&zamow, pozycja);  
apprep(&zamow, pozycja);
```

----- delrep -----

Usunięcie aktualnie wybranego powtórzenia.

```
int delrep(DBF *dbf, unsigned int *rep);
```

Procedura usuwa aktualnie wybrane powtórzenie. Wskaźniki danych powtarzanych są ustawiane na następne powtórzenie. Procedura zwraca 0 gdy brakuje następnego powtórzenia, 1 gdy jest.

Przykład:

Usuwanie pozycji, w których dana "ilosc" ma zerową wartość.

```
DBF zamow;
```

```
int i;
```

```
unsigned int *pozycja;
```

```
float *ilosc;
```

```
...
```

```
do
```

```
{
```

```
    if(*ilosc==0)
```

```
        i=delrep(&zamow,pozycja);
```

```
    else
```

```
        i=nextrep(&zamow,pozycja);
```

```
    }
```

```
while(i);
```

----- newalt -----

Wybranie jednej z opcji wykluczających się.

```
int newalt(DBF *dbf, unsigned char *alt, unsigned int caseno);
```

Danej *alt* określającej numer opcji procedura nadaje wartość *caseno*. Jeżeli z poprzednio wybraną opcją było związane występowanie elementów struktury rekordu (danych), to elementy te są z rekordu usuwane. Elementy struktury związane z obecnie wybraną opcją są wstawiane do rekordu bezpośrednio po danej znakowej określającej numer opcji.

Przykład:

```
#define OPCJA2 2

DBF zamow;

unsigned char *opcjaw;

...

newalt(&zamow, opcjaw, OPCJA2);
```

----- delkon -----

Usunięcie jednej z opcji sumujących się.

```
int delkon(DBF *dbf, unsigned char *kon, unsigned int caseno);
```

Procedura gasi opcję numer *caseno* zmieniając odpowiedni bit w danej *kon* określającej aktualnie wybraną opcję. Jeżeli z gaszoną opcją jest związane występowanie elementów struktury rekordu, to elementy te są z rekordu usuwane.

Przykład:

```
#define OPCJA5 5

DBF zamow;

unsigned int *opcjas;

...

delkon(&zamow, opcjas, OPCJA5);
```

----- addkon -----

Dopisanie jednej z opcji sumujących się.

```
int addkon(DBF *dbf, unsigned char *kon, unsigned char caseno);
```

Procedura zapala opcję numer *caseno* zmieniając odpowiedni bit w danej *kon* określającej aktualnie wybraną opcję. Jeżeli z zapaloną opcją jest związane występowanie elementów struktury rekordu, to elementy te zostaną dopisane do rekordu po danej *kon* i elementach struktury rekordu związanych z zapalonymi opcjami o numerach mniejszych od *caseno*.

Przykład:

```
#define OPCJA6 6
```

```
DBF zamow;
```

```
unsigned int *opcjas;
```

```
...
```

```
addkon(&zamow, opcjas, OPCJA6);
```

----- addtext -----

Wpisanie tekstu do danej tekstowej.

```
int addtext(dbf *dbf, CBTEXT cbtext, char *text);
```

Procedura umożliwia nadawanie nowych wartości danym tekstowym.

Stary tekst będący wartością danej tekstowej *cbtext* jest usuwany z rekordu a nowy tekst przekazywany procedurze przez parametr *text* jest wpisywany do rekordu. W wyniku działania procedury na ogół ulega zmianie długość rekordu.

Przykład:

```
DBF zamow;
```

```
CBTEXT status;
```

```
addtext(&zamow, status, "zrealizowano");
```